

In the Claims:

Please amend the claims as indicated below:

1. (amended) A method embodied in a computer system for monitoring memory usage ~~of~~by an ~~software~~ agent executing in ~~said~~a computer system, wherein said agent is a software agent comprising at least a portion of a software application, said method comprising:

starting a resource tracking ~~application~~function for monitoring memory usage of ~~information associated with said software agent in machine-executable code;~~

creating a computer-readable data structure for storing information about said agent;

identifying a process that is currently running ~~operating~~-on said computer system, and ~~with~~to which said agent is operatively associated;

determining, by said resource tracking application, memory usage data for said agent; and

storing said memory usage data in said data structure;

determining, responsive to said memory usage data in said data structure, that said memory usage of said agent exceeds a predetermined maximum memory usage threshold; and

displaying, responsive to said determination that said memory usage of said agent exceeds said predetermined maximum memory usage threshold, a system administrator user interface, said system administrator user interface including an agent identifier uniquely associated with said agent, a recommended solution to address said exceeding of said predetermined maximum memory usage threshold by said agent, and an execute solution user

interface object, wherein selecting of said execute solution user interface object by a user causes said recommended solution to be automatically performed.

2. (original) The method of claim 1, wherein said computer-readable data structure is a hash table.

3. (currently amended) The method of claim 2, wherein said determining said process that is currently running further comprises determining that said process is identified as being a is a non-hypertext transport protocol (non-HTTP) process comprising at least one of the set consisting of an electronic mail task, a Web conferencing task, and a customer developed task.

4. (currently amended) The method of claim 3 wherein said determining said process that is currently running ~~step~~ further includes:

creating a thread list for monitoring threads associated with said agent;

identifying at least one thread associated with said at least one identified thread to produce at least one determined allocation for said at least one identified thread; and

computing a peak memory usage for said agent using at least one determined allocation;

thereby monitoring memory usage by said ~~a software agent executing in a computer system having a non-HTTP process operating thereon.~~

5. (original) The method of claim 4 further comprising:

comparing said peak usage for said agent to a plurality of peak usages associated with a like plurality of other agents executing in said system.

6. (original) The method of claim 5 further comprising:

computing statistics on said agent and said plurality of other agents; and

ranking said agent against said plurality of other agents based on said peak usage to produce a ranked list.

7. (currently amended) The method of claim 6 further comprising:

displaying said ranked list to ~~said~~ user.

8. (currently amended) The method of claim 2, wherein said determining said process that is currently running further comprises determining that said process is identified as being an HTTP process comprising an HTTP task.

9. (currently amended) The method of claim 8, wherein said determining ~~step~~ said process that is currently running further includes:

identifying HTTP threads operating in said system to produce identified threads, each of said identified threads further having one of a plurality of agent types associated therewith, at least one of said plurality of agent types including said agent, said agent capable of having agent threads associated therewith;

generating an agent thread list for facilitating identification of said plurality of agent types by storing information associated therewith;

identifying which of said plurality of agent types is operating on each of said HTTP threads;

associating those of said agent threads said agent is running on together to produce a related agent set;

determining memory usage for each thread in said related agent set; and

combining said memory usage for each thread in said related agent set to produce a total memory consumption for said agent;

thereby monitoring memory usage by ~~saida software agent executing in a computer system having an HTTP process operating thereon.~~

10. (original) The method of claim 9 further comprising:

comparing said total memory consumption for said agent to a like plurality of total memory consumptions associated with others of said plurality of agents types.

11. (original) The method of claim 10 further comprising:

computing statistics on said total memory consumption for said agent and each one of said plurality of total memory consumptions; and

ranking said agent against said others of said plurality of agent types using said total memory consumption and said plurality of total memory consumptions, respectively, to produce a ranked list.

12. (currently amended) The method of claim 11 further comprising:

displaying said ranked list to ~~saida~~ user.

13. (original) The method of claim 1, further comprising:

determining if said agent is running before determining said memory usage.

14. (original) The method of claim 13, further comprising:

determining if said agent is expired; and

processing said information if said agent is expired.

15. (original) The method of claim 1, wherein said data structure further includes information about a plurality of other software agents.

16. (currently amended) The method of claim 15, further comprising:

establishing ~~said a threshold for~~ maximum memory usage ~~threshold~~.

17. (original) The method of claim 16, further comprising:

terminating said software agent and those of said plurality of other software agents exceeding said threshold.

18. (currently amended) A computer program product including a memory having machine-readable instructions ~~stored~~disposed thereon for, when executed, causing instructing a processor to perform a method for identifying memory usage information associated with a software agent operating in a computer system, said ~~instructions computer program product~~ comprising:

instructions for initiating a resource tracking application ~~function-executable instructions~~ for monitoring memory usage of ~~information associated with~~ said software agent;

instructions for generating a computer-readable data structure residing in computer-accessible memory for storing ~~said~~ memory usage ~~data~~information associated with said agent;

instructions for determining, by said resource tracking application, said memory usage ~~data~~information; and

instructions for storing said memory usage data in said computer-readable data structure;

instructions for determining, responsive to said memory usage data in said data structure, that said memory usage of said agent exceeds a predetermined maximum memory usage threshold; and

instructions for displaying, responsive to said determination that said memory usage of said agent exceeds said predetermined maximum memory usage threshold, a system administrator user interface, said system administrator user interface including an agent identifier uniquely associated with said agent, a recommended solution to address said exceeding of said predetermined maximum memory usage threshold by said agent, and an execute solution user interface object, wherein selecting of said execute solution user interface object by a user causes said recommended solution to be automatically performed.

19. (currently amended) The computer program product of claim 18, said instructions further comprising:

instructions for storing memory usage information about a plurality of other software agents;

instructions for processing said memory usage information associated with said agent and said memory usage information about said plurality of other software agents; and

instructions for generating a rank order list including said information about said agent and said information about said plurality of other software agents.

20. (currently amended) A computer system~~An apparatus~~ for monitoring operation of ~~a software~~ an agent executing in said computer system, wherein said agent is a software agent comprising at least a portion of a software application, said apparatus-computer system comprising:

a processor for executing machine-readable instructions stored in a computer readable memory, said instructions when executed performing the steps of

starting a resource tracking application for monitoring memory usage of said agent;

creating a computer-readable data structure for storing information about said agent;

identifying a process that is currently running on said computer system, and with which said agent is operatively associated;

determining, by said resource tracking application, memory usage data for said agent;

storing said memory usage data in said data structure;

determining, responsive to said memory usage data in said data structure, that said memory usage of said agent exceeds a predetermined maximum memory usage threshold;  
and

displaying, responsive to said determination that said memory usage of said agent exceeds said predetermined maximum memory usage threshold, a system administrator user interface, said system administrator user interface including an agent identifier

uniquely associated with said agent, a recommended solution to address said exceeding of  
said predetermined maximum memory usage threshold by said agent, and an execute  
solution user interface object, wherein selecting of said execute solution user interface  
object by a user causes said recommended solution to be automatically performed~~for~~  
~~starting a resource tracking function for monitoring and processing information~~  
~~associated with a software agent operating in connection with said apparatus; and~~  
~~—— a memory communicatively associated with said processor for storing information~~  
~~about memory usage of said agent in a data structure;~~  
~~—— thereby monitoring said operation of said software agent.~~